IT ŽABLJAK '02

VII

naučno - stručni skup

INFORMACIONE TEHNOLOGIJE

SADAŠNJOST I BUDUĆNOST

Urednici

Novak D. Jauković
Srđan S. Stanković
Srbijanka R. Turajlić

NORMALIZOVANI KORELACIONI KOJEFICIENTI ZA PRETRAŽIVANJE JPEG SLIKA
NORMALIZED CORRELATION COEFFICIENTS FOR SEARCHING JPEG IMAGES

Igor Stojanović, *Customs Administration, Skopje, R. Macedonia*
Dimitar Taškovski, *Faculty of Electrical Engineering Skopje, R. Macedonia*
Ivan Kraljevski, *Veterinary Institute Skopje, Skopje, R. Macedonia*

Sadržaj – *Moguća su dva načina upotrebe normalizovanih korelacionih kojeficienata. Prvi je za lokaciju objekata u JPEG slici, a drugi za lociranje određene slike u biblioteci slika. U članku mi primenjujemo normalizovane korelacione kojeficiente za pretraživanje JPEG slika koristeći progresivnu vejvlet korelaciju zajedno sa Furijeovim metodom. Slika se pretražuje korak po korak. Pretraživanje je sastavljeno od tri inkrementalna koraka i svaki od njih učetvorostručuje broj korelacionih tačaka. Proces se može zaustaviti u svakoj fazi ako međurezultati pokazuju odstupanje.*

Abstract – *There are two ways of using the normalized correlation coefficients. The first one is for locating objects in the JPEG image, and the second one for locating given image in an image library. In the paper we apply normalized correlation coefficients for searching JPEG images. We make use of progressive wavelet correlation along with Fourier method. The image is searched step by step. The searching consists of three incremental steps, each of which quadruples the number of correlation points. The process can be halted at any stage if the intermediate results indicate that the correlation will not result in a match.*

1. INTRODUCTION

A key tool that helped make the Internet universally useful is the text-search engine. The image-search engines available today are relatively crude. There are several techniques for image searching: descriptor-based search, pixel-based search and image understanding techniques. The fastest methods available today use descriptor-based search techniques. IBM QBIC (www.qbic.almaden. ibm.com) [1] is an example of this type of search engine. Images with higher information content, such as satellite images and medical images, are difficult to encapsulate with descriptors. Queries on images of this type require detailed analysis. Normalized correlation coefficients, an instance of pixel-based search techniques, measure the differences between images and patterns. They can be computed with progressive wavelet correlation using Fourier methods [2]. The images are mapped into the wavelet-frequency domain to take advantage of high-speed correlation.

The paper is organized as follows. Section 2 contains the brief description of progressive wavelet correlation using Fourier methods [2]. The results obtained by implementing progressive wavelet correlation using Fourier methods for location of objects in a JPEG image are given in Section 3. Section 4 describes the location of JPEG image in an image library.

2. PROGRESSIVE WAVELET CORRELATION USING FOURIER METHODS

In this section we summarize the technique described in [2]. Progressive wavelet correlation using Fourier methods is based on four theorems: the Wavelet-Correlation Theorem, the Fourier-Wavelet Correlation Theorem, the Fourier-Wavelet Subband Theorem and the Fourier-Wavelet Multiresolution Theorem. To simplify the discussion all data are assumed to be one-dimensional vectors.

The fundamental operation for searching is the circular correlation $\mathbf{x} \otimes \mathbf{y}$. The j^{th} entry of the circular correlation is defined as:

$$(\mathbf{x} \otimes \mathbf{y})_j = \sum_{i=0}^{N-1} x_{i+j \bmod N}\, y_i \,, \quad \text{for } j = 0, 1, \dots, N-1 \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are column vectors of length N.

The matrix form is $\mathbf{x} \otimes \mathbf{y} = \mathbf{X}\mathbf{y}$, where $\mathbf{X}$ is left circulant matrix generated by $\mathbf{x}$ [2], [3].

The notation $(\mathbf{P})_{i\downarrow R}$ denotes subsampling of $\mathbf{P}$ by taking components whose indices are equal to i modulo R. For example, if $R = 4$ and $i = 2$, the subsampling operation extracts the elements 2, 6, 10, 14, ... from $\mathbf{P}$.

Wavelet-Correlation Theorem:

$$(\mathbf{x} \otimes \mathbf{y})_{0\downarrow R} = \sum_{k=0}^{R-1} \left((\mathbf{Hx})_{k\downarrow R}\right) \otimes \left((\mathbf{Hy})_{k\downarrow R}\right) \tag{2}$$

where $\mathbf{H}$ is wavelet-packet transform. $\times$ is the Kronecker product of $\mathbf{I}_M$ and $\mathbf{W}$, $\mathbf{H} = \mathbf{I}_M \times \mathbf{W}$, where $\mathbf{I}_M$ is MxM identity matrix and $\mathbf{W}$ is an RxR matrix with property $\mathbf{W}^T\mathbf{W} = \mathbf{I}_R$. The wavelet transform packet matrix $\mathbf{H}$ has a special structure. $\mathbf{H}$ is block diagonal with block size R. For instance, $\mathbf{W}$ can be 2x2 Haar matrix:

$$\mathbf{W} = \left(\frac{1}{\sqrt{2}}\right)\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{3}$$

Fourier-Wavelet Correlation Theorem:

$$(\mathbf{x}\otimes\mathbf{y})_{0\downarrow R} = \mathbf{F}_M^{-1}\left(\sum_{k=0}^{R-1}(\mathbf{F}_M((\mathbf{Hx})_{k\downarrow R}))*(\hat{\mathbf{F}}_M((\mathbf{Hy})_{k\downarrow R}))\right) \quad (4)$$

where $\mathbf{F}_M$ is the Fourier transform matrix of dimension M and $\hat{\mathbf{F}}_M$ is the complex conjugate of $\mathbf{F}_M$.

Fourier-Wavelet Subband Theorem:

$$\mathbf{F}_N\mathbf{x} = (\mathbf{T}_{N,M,R}\mathbf{H}^{-1})(\mathbf{F}_{M,R}\mathbf{Hx}) \quad (5)$$

where $N = MR$. The matrix $\mathbf{F}_{M,R}$ is an interlaced Fourier transform matrix with structure $\mathbf{F}_{M,R} = \mathbf{F}_M \times \mathbf{I}_R$, that is it has R interlaced copies of transform of size M. The matrix $\mathbf{T}_{N,M,R}$ is a Fourier update matrix that transforms $\mathbf{F}_{M,R}$ into $\mathbf{F}_N$: $\mathbf{F}_N = \mathbf{T}_{N,M,R}\mathbf{F}_{M,R}$.

Fourier-Wavelet Multiresolution Theorem:

$$\mathbf{F}_{N/R,R}\mathbf{H}_1\mathbf{x} = (\mathbf{T}_{N/R,N/R^2,R} \times \mathbf{I}_R)\mathbf{U}_{2,1}^{-1}\mathbf{F}_{N/R^2,R^2}\mathbf{H}_2\mathbf{x} \quad (6)$$

$$\mathbf{F}_N\mathbf{x} = \mathbf{T}_{N,N/R,R}\mathbf{H}_1^{-1}(\mathbf{T}_{N/R,N/R^2,R} \times \mathbf{I}_R)\mathbf{U}_{2,1}^{-1}\mathbf{F}_{N/R^2,R^2}\mathbf{H}_2\mathbf{x} \quad (7)$$

where $N = MR^2$. $\mathbf{H}_2$ is a coarse transform matrix that is block diagonal with block of size R^2 with the structure $\mathbf{H}_2 = \mathbf{I}_{N/R^2} \times (\mathbf{W}_1 \times \mathbf{W}_1)$ and operates on R^2 subbands, each of length N/R^2. $\mathbf{W}_1$ is an RxR wavelet filter matrix with property $\mathbf{W}_1^T\mathbf{W}_1 = \mathbf{I}_R$. $\mathbf{H}_1$ is a fine transform matrix that is block diagonal with block size N/R with structure $\mathbf{H}_1 = \mathbf{I}_{N/R} \times \mathbf{W}_1$. There is an update matrix $\mathbf{U}_{2,1}$ that refines $\mathbf{H}_1$ into $\mathbf{H}_2$, $\mathbf{H}_2=\mathbf{U}_{2,1}\mathbf{H}_1$. The matrix $\mathbf{U}_{2,1}$ is block diagonal with block of size R^2 with the following structure:

$$\mathbf{U}_{2,1} = \mathbf{I}_{N/R^2} \times (\mathbf{W}_1 \times \mathbf{I}_R) \quad (8)$$

3. LOCATION OF OBJECTS IN A JPEG IMAGE

JPEG compression is based on the discrete-cosine transform (DCT) [4]. The matrix $\mathbf{C}_8$ is an 8x8 DCT matrix that is used to create transforms of 8x8 subimages in a JPEG representation of an image. The multiresolution search process relies on the factorization of the DCT matrix $\mathbf{C}_8 = \mathbf{V}_{8,4}\mathbf{V}_{4,2}\mathbf{V}_2$, where $\mathbf{V}_2$ and $\mathbf{V}_{4,2}$ are matrices built from Kronecker products of $\mathbf{W}$ and the identity matrix. The matrix $\mathbf{V}_2 = \mathbf{I}_4 \times \mathbf{W}$ consists of 4 interlaced copies of $\mathbf{W}$ and is of size 8x8. The matrix $\mathbf{V}_{4,2}$ has a structure $\mathbf{V}_{4,2} = \mathbf{I}_2 \times (\mathbf{W} \times \mathbf{I}_2)$. If we write $\mathbf{C}_8 = \mathbf{V}_{8,4}\mathbf{V}_{4,2}\mathbf{V}_2$ where $\mathbf{V}_{8,4}$ is a matrix whose coefficients we want to compute, then

$$\mathbf{V}_{8,4} = \mathbf{C}_8\mathbf{V}_2^{-1}\mathbf{V}_{4,2}^{-1} \quad (9)$$

obtain the last expression by multiplying both sides by $\mathbf{V}_2^{-1}\,\mathbf{V}_{4,2}^{-1}$. The matrix $\mathbf{V}_{8,4}$ satisfies equation $\mathbf{V}_{8,4} = \mathbf{V}(\mathbf{W} \times \mathbf{I}_4)$. A formula for the inverse of V can be found in [2, 3].

The matrix $\mathbf{H}$ is an NxN matrix with the structure $\mathbf{I}_M \times \mathbf{C}_8$ where $N = 8M$. It produces the JPEG transform of a vectors of length N. Let $\mathbf{x}$ be image stored as a JPEG transform of a vector $\mathbf{Hx}$ with an instance of a pattern $\mathbf{y}$ with JPEG transform $\mathbf{Hy}$. To locate object $\mathbf{y}$ in a JPEG image we search the image step by step. The algorithm consists of three incremental steps, each of which quadruples the number of correlation points. The process can be halted at any stage if the intermediate results indicate that the correlation will not result in a match.

The three incremental steps are:

1. *Coarse correlation* – Generate the Fourier transforms $\mathbf{F}_{M,8}\mathbf{Hx}$ and $\hat{\mathbf{F}}_{M,8}\mathbf{Hy}$. Multiply the transforms point by point and partition them into eight subbands of length M. Add these eights vectors and take the inverse Fourier transform of the sum. Every eighth point of the correlation is generated.

2. *Medium correlation* – Multiply $\mathbf{F}_{M,8}\mathbf{Hx}$ by $(\mathbf{T}_{2M,M,2} \times \mathbf{I}_4)(\mathbf{I}_M \times ((\mathbf{W} \times \mathbf{I}_4)\mathbf{V}^{-1}))$ and $\hat{\mathbf{F}}_{M,8}\mathbf{Hy}$ by $(\hat{\mathbf{T}}_{2M,M,2} \times \mathbf{I}_4)(\mathbf{I}_M \times ((\mathbf{W} \times \mathbf{I}_4)\mathbf{V}^{-1}))$. Multiply the resulting vectors point by point and partition them in four subband of length $2M$. Add the subbands and you will create a single vector of length $2M$. Taking the inverse Fourier transform of size $2M$ yields the correlation at indices that are multiples of 4 mod 8 of the full correlation.

3. *Fine correlation* – Multiply the $\mathbf{x}$ and $\mathbf{y}$ transform from the preceding step by $(\mathbf{T}_{4M,2M,2} \times \mathbf{I}_2)$ $(\mathbf{I}_M \times \mathbf{V}_{4,2})$ and $(\hat{\mathbf{T}}_{4M,2M,2} \times \mathbf{I}_2)(\mathbf{I}_M \times \mathbf{V}_{4,2})$, respectively. Multiply the resulting vectors point by point and partition them in two subbands of length $4M$. Add the subbands and you will create a single vector of length $4M$. Take the inverse Fourier transform of size $4M$ to obtain the correlation at indices that are multiples of 2 mod 8 and 6 mod 8 of the full correlation.

4. *Full correlation* – Multiply the $\mathbf{x}$ and $\mathbf{y}$ transform from the last step by $\mathbf{T}_{8M,4M,2}(\mathbf{I}_M \times \mathbf{V}_2)$ and $\hat{\mathbf{T}}_{8M,4M,2}(\mathbf{I}_M \times \mathbf{V}_2)$, respectively. Multiply the resulting vectors point by point and take the inverse Fourier transform of size $8M$ to obtain the correlation at odd indices.

We investigate what happens in the two-dimensional case. The assumption is that the image size is N by N. In step 1, we have 64 subbands of length $N^2/64$. We perform one step of the inverse 2D $\mathbf{H}$ function, and one 2D step of the forward Fourier transform function. Fortunately, these steps are simple generalizations of the 1D functions. Specifically, if in 1D we compute $\mathbf{Hx}$, where $\mathbf{H}$ is N by N and $\mathbf{x}$ is N by 1, then in 2D we compute **H2X** conjtrans (**H2**), where $\mathbf{X}$ is the 2D image to be transformed, **H2** is N by N, and each row of **H2** is equal to $\mathbf{H}$. This is equivalent to applying $\mathbf{H}$ to each of the columns of $\mathbf{X}$ and $\mathbf{H}$ to each of the rows of $\mathbf{X}$. The next step is to add the 64 subbands point by point to create a 2D array of size $N/8$ by $N/8$. If we take its inverse Fourier transform, we will obtain the correlations at points that lie on a grid that is coarser than the original pixel grid by a factor 8 in each dimension.

In step 2, we obtain 16 subbands of size $N^2/16$ by adding the 16 subbands point by point, and taking the Fourier inverse. We will obtain the correlation values on a grid that is coarser than the original grid by a factor of 4 in each dimension.

In step 3, we obtain 4 subbands of size $N^2/4$. In step 4, we obtain the full resolution.

Formulas for calculating of normalized correlation coefficients for measuring differences between images and patterns are given in [2]. Normalized correlation coefficients can be computed from the correlations described above. The normalization is very important because it allows for a threshold to be set. Such a threshold is independent of the encoding of the images.

The normalized correlation coefficient has a maximum absolute value of 1. Correlations that have absolute values above 0.9 are excellent, and almost always indicate a match found. Correlations of 0.7 are good matches. Correlations of 0.5 are usually fair or poor. Correlations of 0.3 or less are very poor. There is a tradeoff between the value of the threshold and the likelihood of finding a relevant match. Higher thresholds reduce the probability of finding something that is of interest, but they also reduce the probability of falsely matching something that is not of interest.

An example of locating object in a JPEG image and conclusions are given in [3].

4. LOCATING A JPEG IMAGE IN THE IMAGE LIBRARY

The normalized correlation coefficients are impractical to use during image processing owing to the big number of operations per image they entail. Current processor performance does not allow a practical implementation of this method. One alternative is to combine IBM QBIC [1] and the normalized correlation coefficients. This yields an improvement in the locating time for a given JPEG image in the library.

The QBIC technology allows content-based queries. More precisely, it performs queries based on visual properties of the input image, such as color or texture.

Querying by content requires two phases:

- *Database creation*: A preprocessing step to compute numeric features for an image or set of images, and store them in a database.
- *Database query*: The run-time step that uses the computed features to find images similar to a query specification.

The database creation step uses one or more feature classes to compute the features of input images as numeric values. Each feature class creates a feature table in the database where these computed values are stored. During the database query phase, QBIC compares feature data in the query with the computed data in the feature tables. A query can search on one or more features for similarity.

A simple query involves only one feature. An example of a simple query would be to find images in the database that have a color distribution similar to the query image. A complex query involves more than one feature and can be either a multi-feature or a multi-pass query. In a multi-feature query, QBIC searches through different types of feature data in the database in order to find images that closely resemble the query image. All feature classes are treated equally during the database search, and all involved feature tables are searched at the same time. An example of a multi-feature query would be finding images in the database that have a color distribution and texture similar to a query image.

In a multi-pass query, the output of an initial search is used as the input for the next search. QBIC reorders the search results from a previous pass based on the feature distances in the current pass. An example of a multi-pass query would be finding images in the database that have a color distribution similar to the query image, and then reordering the results based on color composition.

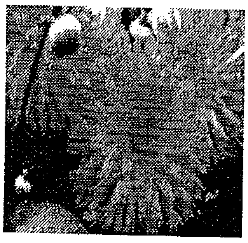
For multi-feature and multi-pass queries, features can be weighted to specify their relative importance, which provides flexibility for advanced applications where the returned results must be fine-tuned.

QBIC queries may be of one of the following types:

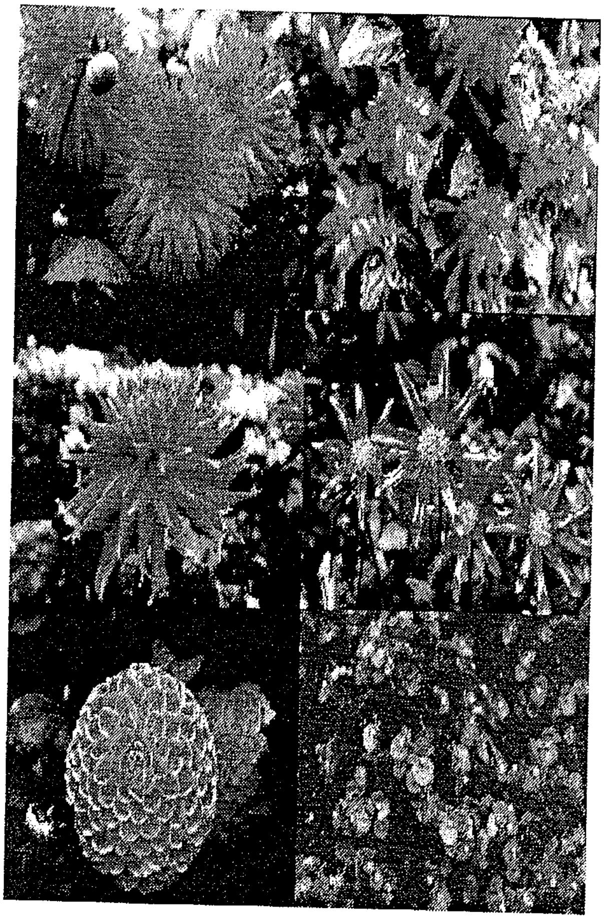
1. *Query By Example*: This type of query locates images in a database that are similar to the "example" query image (whose feature data has already been computed by QBIC).

2. *Query By Image*: This type of query may be invoked on images whose feature data has not been precomputed by QBIC. QBIC computes the feature data for the input image, and then compares it with data in database.

Using QBIC, we established a database for two characteristics of JPEG images: color histogram and texture feature. We query the database for those images in the library with the most similar characteristics to the input image. On this set of candidates, we apply the normalized correlation coefficients to obtain the desired image. As an example, we show our work on locating the image flower02.jpg (Pic. 2) in a database. Using QBIC, we isolated six candidate images based on the Color Histogram Feature (Pic. 3).



Pic. 2. Flower02.jpg



Pic. 3. Candidate images

5. CONCLUSIONS

This paper proposes a procedure for searching JPEG images using normalized correlation coefficients. Our experimental results confirm that, generally, the procedure can be applied successfully for this type of search.

The principal difficulty arises from low processing speed, which hinders this approach from becoming commercially viable at the moment. However, as processing speed increases in the next few years, we expect the procedure to become useful in practice for detail analysis at rates of up to 1000 images per second.

With higher processing speed, it would be possible to construct a system that combines search by description with search by pixels. The descriptor could be used to isolate a chunk out of a big collection of images, while the pixel based search engine would look for a detailed object in this chunk of images.

REFERENCES

[1] M. Flickner et al., "Query by image and video content: The QBIC system," *IEEE Comp.*, vol. 28, pp. 23-32, Sept. 1995.

[2] H. S. Stone, "Progressive Wavelet Correlation Using Fourier Methods," *IEEE Trans. Signal Processing*, vol. 47, pp. 97-107, Jan. 1999.

[3] I. Stojanovic, M. Bogdanov, "Location of Objects in a JPEG Image with Progressive Wavelet Correlation using Fourier methods", *9th Telecommunications Forum*, pp. 561-564, Belgrade, Yugoslavia, Nov. 20-22, 2001.

[4] G. K. Wallace, "The JPEG still-picture compression standard", *Commun. ACM*, vol. 34, no.4, pp. 30-44, Apr. 1991.